

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 29 (2015) 698 – 703

www.elsevier.com/locate/procedia

The 22nd CIRP conference on Life Cycle Engineering

Two-stage sequence generation for partial disassembly of products with sequence dependent task times

Robert J. Riggs^{a,b,*}, Xiaoning Jin^c, S. Jack Hu^{b,c}^aDepartment of Industrial Engineering, Clemson University, Clemson, SC 29634, USA^bDepartment of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI 48109, USA^cDepartment of Mechanical Engineering, The University of Michigan, Ann Arbor, MI 48109, USA* Corresponding author. E-mail address: robriggs@umich.edu

Abstract

For disassembly sequence generation, partial disassembly and sequence dependent task times are typically not considered together in the same model. We developed a two-stage optimization program that first determines the optimal partial disassembly sequence according to reuse value only, followed by the second stage that finds the optimal partial disassembly sequence that also includes sequence dependence task times. We prove the optimality of the two-stage approach under the condition that all components with any positive reuse value must be included in the final sequence. If this condition does not need to be met, a task hedging policy is shown to be effective.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the scientific committee of The 22nd CIRP conference on Life Cycle Engineering

Keywords: disassembly sequence generation; end of life (EOL); partial disassembly; sequence dependent task times

1. Introduction

Remanufacturing is a promising and responsible process for dealing with end of life (EOL) products. Instead of simply disposing such products in landfills or dismantle them for materials recycling, remanufacturing recovers value in products through sorting, cleaning, and disassembling into modules or components, so that components or modules are reused directly or reconditioned. Disassembly is the backbone of the remanufacturing process because this is the stage where the end of life value of components and modules can be realized. Only certain components and modules contain reuse value when the core reaches its EOL, but this value has to be mined through disassembly.

Disassembly is typically a labor-intensive manual process and there is also a lot of task variation associated with the disassembly of cores. The task time to disassemble each component is typically a random number following certain probability distribution since the work is manually done and the condition of the cores can vary from very good to bad. There is also variation in the actual EOL condition of each component. Components can be missing, damaged, either

have reuse value or no reuse value depending on the damage to it, and these combination of EOL conditions creates many possible EOL states that the core can be in. In addition to the variation of EOL states, the actual disassembly process can be quite harsh and damaging to the core's components [1], [2]. "Disassembly damage" describes the process where aged parts in a core of the returned products are likely to break during the disassembly process [3]. Damage can occur because of gravity where parts fall due to the orientation during disassembly, or there is improper handling either due to operator error or during a long and difficult disassembly process.

One method to help reduce the possibility of disassembly damage is the design of fixtures [1], but it can be difficult to design a fixture for all the possible orientations a core might need to be in during the disassembly process. Another idea is to extract the most valuable components and modules from the core as early as possible in the disassembly sequence to prevent the possibility of damage to the most valuable components or modules. This is one of the primary goals of this paper, to create a disassembly sequence model that seeks

to extract the most valuable components and modules as early in the process as possible.

For disassembly sequence generation, quite often the end point of the disassembly sequence is either complete or selective, meaning a pre-determined end point is chosen before the optimal sequence is determined. We first consider partial disassembly for our model, where the mathematical program determines the best disassembly stopping point in order to maximize the retained value from the core. Then a disassembly sequence model is developed to consider the possibility of sequence dependent task times. Sequence dependent task times refer to the relationship certain components have with one another based on the order they are removed from the core. For example, if component A is removed before component B, then the removal time of component B may decrease accordingly; however, if component A is removed after component B, then the removal time of component B remains the same. Depending on the removal order of components, the overall disassembly time can decrease.

Kang et al. [4] developed a disassembly sequencing method for sequence dependent operation times with the objective of profit maximization. Scholl et al. [5] considered sequence dependence task times but for the application of assembly line balancing. Kalayci and Gupta [6] used an artificial bee colony optimization technique to solve the disassembly line balancing problem when there is sequence dependence.

Altekin et al. [7] developed an exact Mixed Integer Programming (MIP) formulation for solving a partial disassembly line balancing problem. Tripathi et al. [8] proposed a disassembly optimization model that determines the disassembly sequence and the depth of disassembly to maximize revenue. Rickli et al. [9] developed a multi-objective genetic algorithm for partial disassembly sequence generation. Bentaha et al. [10] developed a stochastic line balancing method to create a disassembly sequence where partial disassembly is considered. Their objective is to maximize profit under uncertainty with task times. Rickli et al. [11] developed an approach for partial disassembly sequence generation where the EOL product quality is uncertain. Kara et al. [15] created a selective disassembly method based on a method developed by Nevins and Whitney for assembly that creates a graphical representation of the disassembly sequence.

The remainder of the paper is organized as follows: Section 2 presents the two-stage sequence generation model and the heuristic policy. Section 3 provides a simple case study to show the effectiveness of our two-stage model, Section 4 discusses the results, and Section 5 concludes the paper and summarizes possible future work.

2. Method

We assume that the EOL condition possibilities for each component or set of components are known and that the probability of that condition is also known; therefore, the EOL state probabilities are also known.

The notation for each model is the following:

Indexes:

- i for the component number ($1-N$);

- j for the EOL state;
- k for the position in the sequence ($1-K$).

Parameters:

- N for total set of components to be disassembled ($i = 1, \dots, |N|$);
- J for the total number of EOL states ($j = 1, \dots, |J|$);
- K for the total number of positions in the sequence ($k = 1, \dots, |K|$);
- v_{ij} for the profit/value of component i in EOL state j ;
- p_j for the probability of EOL state j ;
- w for the weight on the objective;
- $P(i)$ for the immediate successor set for task i .

Variables:

- $x_{ik} = 1$, if component i is removed in the sequence at position k , 0 otherwise;
- $z_i \in (1, \dots, K)$, component i is assigned to the $(1, \dots, K)$ position in the sequence.

A two-stage approach is used because the methods to model partial disassembly and sequence dependent task times have conflict when done simultaneously. Partial disassembly allows for the removal of certain components to not be a part of the final disassembly sequence. The method to model sequence dependent task times requires every disassembly task that is a part of the original group of $|N|$ to be assigned a position in the sequence to assess if certain time savings will result based on the final chosen disassembly sequence.

The same result can be achieved using a two-stage approach where the first stage determines which components need to be hedged from the disassembly sequence solely based on reuse value. This new set of components will be passed to the second stage of the method and the disassembly sequence reordered with respect to sequence dependent task times.

2.1 First stage: partial disassembly

The first stage method for partial disassembly can also be referred to as a “greedy” model because it orders the disassembly sequence solely based on the reuse value; therefore, components with higher reuse value will be pushed towards the front of the sequence, while components with less reuse value will be pushed towards the end of the sequence.

The objective function (1a) maximizes the total EOL reuse value for all EOL states j .

$$\text{maximize } Y = \sum_{i=1}^N \sum_{j=1}^J \sum_{k=1}^K p_j \left(x_{ik} v_{ij} - \left(w v_{ij} \left(\frac{k x_{ik}}{K} \right) \right) \right) \quad (1a)$$

The objective function (1a) can be re-written as (1b).

$$\text{maximize } Y = \sum_{i=1}^N \sum_{j=1}^J \sum_{k=1}^K p_j v_{ij} \left(x_{ik} - w \left(\frac{k x_{ik}}{K} \right) \right) \quad (1b)$$

For the objective function (1b), the value v_{ij} is weighted by the term p_j , the probability of each EOL state j . The position the component i is removed in the sequence z_i , which goes from 1 to K , is divided by K , the total number of spots in

the sequence, making it a fractional number or equal to 1. This term is weighted by the value of w and subtracted from the binary variable x_{ik} , which is 1 if component i is in position k of the sequence. The constraints for the mixed integer program are the following:

$$x_{ik} \in \{0,1\} \forall i \in N \text{ and } k \in K \quad (2)$$

$$\sum_{k=1}^K x_{ik} = 1 \forall i \in N \quad (3)$$

$$\sum_{i=1}^N x_{ik} = 1 \forall k \in K \quad (4)$$

$$\sum_{k=1}^K x_{ik} k = z_i \forall i \in N \quad (5)$$

$$z_i \in \{1,2, \dots, K\} \forall i \in N \quad (6)$$

$$z_i < z_{\hat{i}} \forall i, \hat{i} \in N, i \neq \hat{i}, \hat{i} \in P(i) \quad (7)$$

Constraint (2) ensures x_{ik} is a binary integer variable. Constraint (3) ensures that task i holds only one position in the sequence. Constraint (4) guarantees that each position in the sequence only contains one task i . Constraint (5) defines the value of z_i . Constraint (6) ensures z_i is an integer in the range of 1 to K . Constraint (7) are the precedence constraints.

The first stage problem is run for a given problem and a partial disassembly sequence is produced. The stage one optimization program pushes the components with the highest reuse value as far to the front of the sequence as possible. As long as every component does not have reuse value, there will likely be a position in the sequence, called position Ω , where the component in that sequence position has reuse value and every component removed after that position does not have reuse value. Each component removed after the Ω position will be hedged from the component pool that will be passed to the second stage problem to determine the disassembly sequence order with the presence of sequence dependent task times.

2.2 Second stage: sequence dependent task times

In the second stage, only the components not hedged from the sequence will be used as inputs for the final disassembly sequence. The time to remove components will now be a part of the objective function, where it is more costly to have a longer disassembly sequence. Additional parameters and variables are required for the second stage problem as shown below.

Parameters:

- sd_{ijm} for the sequence dependent time difference if component i is removed before component m in state j
- c_i for the unit time cost for component i
- t_{ij} for the task time for component i in state j

Variables:

- $y_{im} = 1$ if component i is removed before component m , 0 otherwise.

Parameter sd_{ijm} can be either positive or negative, depending on whether removing component i before component m in the sequence saves total removal time or not.

The parameter c_i is the cost per unit time to remove component i in the sequence and the task time for each component in each state j is t_{ij} . An additional decision variable is needed to determine if component i is removed before component m or not. This is a binary variable and if true then it is a 1 and 0 otherwise.

The objective function for the second stage problem is shown in (1c) below.

$$\begin{aligned} \text{maximize } Y = & \left(\sum_{i=1}^N \sum_{j=1}^J \sum_{k=1}^K p_j \left(x_{ik} v_{ij} - \left(w v_{ij} \left(\frac{k x_{ik}}{K} \right) \right) \right) \right) - \\ & \sum_{i=1}^N \sum_{j=1}^J \sum_{k=1}^K p_j c_i x_{ik} t_{ij} - \\ & \sum_{\substack{i,m=1 \\ i \neq m}}^N \sum_{j=1}^J \sum_{k=1}^K p_j c_i s d_{ijm} y_{im} \end{aligned} \quad (1c)$$

The objective function (1c) has three main parts. The first part is very similar to objective function (1a), where the EOL value is weighted and multiplied by the ratio of the components removal position in the sequence and subtracted from the initial reuse value of that component in EOL state j . This first part of the objective function will still push the components with the highest reuse value towards the front of the sequence. The second part of objective function (1c) takes into account the task time for each component i in EOL state j . The total task time of the chosen disassembly sequence is multiplied by c_i , the unit time removal cost of component i . The third part of (1c) is where the sequence dependent task times are brought into the equation. If component i is removed before component m , and there is a difference in removal time for one of the components because i is removed before m , then sd_{ijm} will take on a value other than zero. If the difference in removal time is negative, meaning there is a time savings based on removal order, then the third part of (1c) will take on a positive value since the objective is maximization.

The second stage problem will have objective function (1c) and be subject to the previous constraints (2), (3), (4), (5), (6), and (7), with the addition of the constraints (8)-(9) described below.

Constraints (8) and (9) determine which components/modules are removed before one another.

$$z_i - z_m \leq M(1 - y_{im}) \forall i, m \in N, i \neq m \quad (8)$$

$$z_m - z_i \leq M y_{im} \forall i, m \in N, i \neq m \quad (9)$$

Constraint (10) ensures that y_{im} is integer and binary.

$$y_{im} \in \{0,1\} \forall i, m \in N \quad (10)$$

Constraints (11) and (12) guarantees y_{im} and y_{mi} are not equivalent.

$$y_{im} \neq y_{mi} \forall i, m \in N, i \neq m \quad (11)$$

$$y_{im} + y_{mi} = 1 \forall i, m \in N, i \neq m \quad (12)$$

Constraints (8) – (12) link each of the decision variables to one another.

2.3 Proof of optimality for the two-stage problem

Proposition 1: *The final disassembly sequence solution using the two-stage approach is guaranteed to be at least as good as any other disassembly sequence solution to the problem under the condition that every component that has some EOL value must be included in the final sequence and the components hedged from the sequence do not have sequence dependent task times.*

Proof: The proof for this proposition is intuitive. If given the condition that every component that has any EOL value must be included in the final disassembly sequence and only components were hedged from the sequence without sequence dependent task times, then only components that have no EOL value will be removed from the sequence. Stage 1 of the two-stage approach only removes components from the sequence with no EOL value, and all component removal tasks are hedged from the sequence at the point when (working backwards from the end of the sequence) the first component has EOL value.

In addition, the second stage will find an optimal sequence only using components that are required to be included in the final disassembly sequence and will include sequence dependent task time criteria. □

2.4 Heuristic Two-stage Policy

Under the condition that not every component with any EOL value must be included in the final disassembly sequence, the two-stage approach is no longer guaranteed optimal and a heuristic approach is developed. The heuristic approach is the same for the first stage, all components that have no EOL value are pushed to the end of the sequence and are hedged after the position in the sequence where the final component that does have EOL value is assigned. An example of this is shown in Fig. 1. In position 8, component B is the final component in the sequence that has an EOL reuse value, so all components after position 8 (positions 9 and 10) are removed from the sequence, and all components earlier in the sequence will remain in the sequence for now.

Pos. In Seq.	...	7	8	9	10
Component	...	A	B	C	D
EOL Value	...	\$4.00	\$1.25	\$-	\$-

← Remain
→ Hedged

Fig. 1 first stage hedging example

For the second part of the first stage, a stopping point in the sequence needs to be determined, i.e., finding a point in the sequence after which it is no longer justified financially to remove a component. To do so, we work our way from the back of the sequence. Based on the example in Fig. 1, first component B will be assessed in position 8. If the weighted EOL value of component B for all EOL states is greater than or equal to the total cost to remove component B, then component B will remain in the sequence. If it is more expensive to remove component B than the weighted EOL

value, then component B will be hedged from the sequence and we will assess the next component in the sequence in position 7. The previously described assessment is shown in equations (13) and (14).

$$\sum_{j=1}^J p_j v_{ij} \geq \sum_{j=1}^J p_j t_{ij} c_i \quad (13)$$

$$\sum_{j=1}^J p_j v_{ij} < \sum_{j=1}^J p_j t_{ij} c_i \quad (14)$$

If equation (13) holds, then the component remains in the sequence, no other component needs to be assessed, and the process moves on to the second stage of the two-stage approach. If equation (14) holds, then the component being assessed needs to be hedged from the sequence and the assessment will continue until a stopping point is found, and then the second stage can commence.

3. Case Study

To demonstrate the effectiveness and procedure of the two-stage algorithm, we will use the disassembly of a laptop as an example. The laptop case study is simplified for ease of explanation. This laptop case study has been used in [12], [13], and [14].

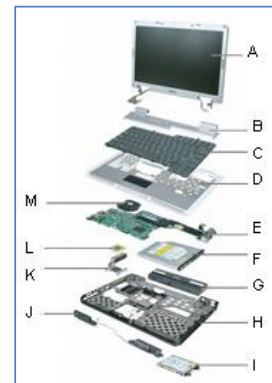


Fig. 2. Exploded view of laptop, [12]

Table 1. Laptop bill of materials, [12]

A	display assembly
B	hinge cover
C	keyboard
D	palm rest (with touch pad)
E	system board
F	optical drive
G	main battery
H	computer base
I	hard drive
J	speakers
K	microprocessor assembly
L	microprocessor
M	fan

Figure 2 shows the exploded view of the simplified laptop case study and table 1 shows the laptop bill of materials. The precedence graph for the disassembly of the laptop is shown in figure 3.

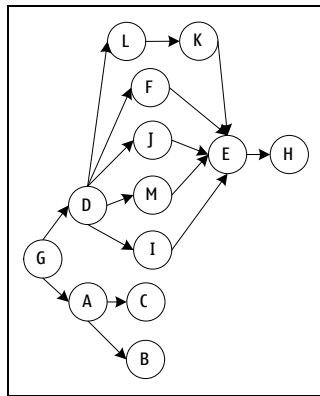


Fig. 3. Laptop precedence graph

The two-stage algorithm can model different EOL states, so for the case study we have 3 EOL conditions shown below.

- 1) Condition 1: There is a 65% probability that the hard-drive (component *I*) is present in the assembly and a 35% chance the hard-drive is missing, so zero task time if missing.
- 2) Condition 2: The system board (component *E*) can be damaged with a 35% probability and if damaged, will have no reuse value.
- 3) Condition 3: The optical drive (component *F*) has a 25% probability of being more difficult to remove than normal and results in a longer standard task time, 5 time units instead of 3 time units.

The EOL conditions can be summarized by 8 total EOL states, shown in table 2.

Table 2. Probabilities for each EOL state

EOL State	EOL Condition			EOL State Probability
	I	E	F	
1	0.65	0.65	0.75	0.317
2	0.65	0.65	0.25	0.106
3	0.65	0.35	0.75	0.171
4	0.65	0.35	0.25	0.057
5	0.35	0.65	0.75	0.171
6	0.35	0.65	0.25	0.057
7	0.35	0.35	0.75	0.092
8	0.35	0.35	0.25	0.031

The probability of each EOL state is found by multiplying the probabilities of each condition in that state. For example, EOL state 1 has the hard-drive present, the system board not damaged, and the optical drive is not more difficult to remove than normal ($0.65 \times 0.65 \times 0.75 = 0.317$). Each EOL state has a different combination of each EOL condition possibility. The EOL state probabilities will be the p_j values that are used in each sequence generation model.

Each component has a certain value at its EOL, either through direct reuse or recycling. Some components do not have any tangible reuse value, or it might cost more money to recycle the component than it is worth. Table 3 contains the EOL values for each component. It is worth noting that these EOL values are only for components that are present in the assembly and not damaged.

Table 3. EOL value for each laptop component

Component	EOL Value
A display assembly	\$ 16.00
B hinge cover	\$ -
C keyboard	\$ -
D palm rest (with touch pad)	\$ -
E system board	\$ 12.00
F optical drive	\$ 7.00
G main battery	\$ 8.00
H computer base	\$ 2.50
I hard drive	\$ 11.00
J speakers	\$ -
K microprocessor thermal-cooling assembly	\$ -
L microprocessor	\$ -
M fan	\$ 1.50

3.1 First stage: partial disassembly

Using the data from the figures and tables in section 3, the first stage of the algorithm will have objective (1b) and will be subject to constraints (2)-(7). Table 4 contains the results for the disassembly sequence that will be hedged. The value of w used for the first-stage problem is 0.5 and the objective is 41.64. This objective value comes from running the math program in section 2.1.

Table 4. Partial disassembly sequence order

Position	Component	EOL Value
1	G	\$ 8.00
2	A	\$ 16.00
3	D	\$ -
4	F	\$ 11.00
5	I	\$ 7.00
6	M	\$ 1.50
7	J	\$ -
8	L	\$ -
9	K	\$ -
10	E	\$ 12.00
11	H	\$ 2.50
12	C	\$ -
13	B	\$ -

Highlighted in grey at the 11th position is component *H*, which is in the Ω position. Every component in the sequence after *H* (*C* and *B*) will be hedged from the sequence and no longer removed from the core. Only the removal of components in positions 1-11 will be passed to the second-stage for consideration in the second-stage.

3.2 Second stage: sequence dependent task times

For the second stage problem, the objective will be (1c) subject to constraints (2)-(12). Task time is now included in the objective, and the removal time for each component passed to the second-stage problem is shown in table 5. The parameter c_i is assumed to be the same for the removal of all

components i and is equal to 0.5. It is determined that if component J is removed after component L , then there will be a time savings of 1 time unit for the removal of component J due to sequence dependence.

Table 5. Task time for each component

Component	Task Time (tu)
A	4
D	5
E	5
F	3
G	2
H	2
I	6
J	5
K	3
L	1
M	2

The final sequence after the second stage problem is the removal of components $G, A, D, F, I, M, L, K, J, E$, and H in that order. The objective of the second stage problem when running the math program from section 2.2 is 31.93.

4. Discussion

The sequence order for the first stage and second stage problems are very similar. The differences are components B and C are hedged from the sequence in the second stage, and the first stage problem has the sequence order of J, L , and K , while the second stage sequence has the order L, K , and J . The sequences for both stages are the similar because the objective functions (1b) and (1c) will order the components in a “greedy” fashion such that components with higher EOL value will be pushed towards the front of the sequence. The second stage problem considers sequence dependence and task time in the objective, and since the removal of component J after L results in a time savings, the objective (1c) is maximized with J being removed after L in the sequence.

5. Conclusions and Future Work

In this paper we presented a two-stage sequence generation model for product disassembly in a remanufacturing system. The first stage ordered the removal of components in the sequence with respect to reuse value where the most valuable components were pushed to the front of the sequence and components with no reuse value were pushed to the end of the sequence. The components with no reuse value were hedged from the sequence to create a partial disassembly problem. The second stage of the problem focused on a model for sequence dependent task times, where the order of the sequence was influenced not only by components with higher reuse value to be pushed towards the start of the sequence but saving time during the disassembly can impact the final order of the sequence. We proved the optimality of the two-stage approach under certain conditions and if these conditions do

not hold then a two-stage heuristic policy can be implemented. This two-stage model considers both partial disassembly and sequence dependent task times, which have not been considered in the existing disassembly sequencing literature.

Our future work should focus on addressing the limitations of the two-stage model. For a product with very few components to consider for the disassembly sequence, each stage of the model has a short computation time; however, for larger instances of the problem, having to run two binary programs to come to a final solution may be too time consuming and not efficient. Eliminating the need for a two-stage problem and reducing this model to a single model can be an important improvement. Lastly, we assume that all EOL conditions are known and the percentage for each EOL condition is also assumed. This assumption can be further released to the case that the EOL conditions are not completely known for practical applications of disassembly.

References

- [1] Jovane F, Altling L, Armillotta A, Eversheim W, Feldmann K, Seliger G, Roth N. A key issue in product life cycle: disassembly. *CIRP Annals-Manufacturing Technology* 1993;42(2):651-658.
- [2] Bras B, Hammond R. Towards design for remanufacturing-metrics for assessing remanufacturing. *Proceedings of the 1st International Workshop on Reuse* 1996:5-22.
- [3] Gungor A, Gupta SM. Disassembly sequence planning for products with defective parts in product recovery. *Computers and Industrial Engineering* 1998;35(1-2):161-164.
- [4] Kang JG, Lee DH, Xirouchakis P, Persson JG. Parallel disassembly sequencing with sequence-dependent operations times. *CIRP Annals-Manufacturing Technology* 2001;50(1):343-346.
- [5] Scholl A, Boysen N, Fließner M. The sequence-dependent assembly line balancing problem. *OR Spectrum* 2008;30:579-609.
- [6] Kalayci CB, Gupta SM. Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem. *Expert Systems with Applications* 2013;40(18):7231-7241.
- [7] Altekun FT, Kandiller L, Ozdemirel NE. Profit-oriented disassembly-line balancing. *International Journal of Production Research* 2008;46(10):2675-2693.
- [8] Tripathi M, Agrawal S, Pandey MK, Shankar R, Tiwari MK. Real world disassembly modeling and sequencing problem: optimization by algorithm of self-guided ants (ASGA). *Robotics and Computer-Integrated Manufacturing* 2009;25(3):483-496.
- [9] Rickli JL, Camelio JA. Multi-objective partial disassembly optimization based on sequence feasibility. *Journal of Manufacturing Systems* 2013;32(1):281-293.
- [10] Bentaha ML, Battaia O, Dolgui A. Disassembly line balancing and sequencing under uncertainty. *Procedia CIRP* 2014;15:239-244.
- [11] Rickli JL, Camelio JA. Partial disassembly sequencing considering acquired end-of-life product age distributions. *International Journal of Production Research* 2014;52(24):7496-7512.
- [12] Hu SJ, Ko J, Weyand L, ElMaraghy HA, Lien TK, Koren Y, Bley H, Chrysosolouris G, Nasr N, Shpitalni M. Assembly system design and operations for product variety. *CIRP Annals-Manufacturing Technology* 2011;60:715-733.
- [13] Riggs RJ, Hu SJ. Disassembly liaison graphs inspired by word clouds. *Procedia CIRP* 2013;7:521-526.
- [14] Riggs RJ, Jin X, Hu SJ. Disassembly line balancing under high variety of end of life states using a joint precedence graph approach. *Journal of Manufacturing Systems* 2015; In the Press.
- [15] Kara S, Pornprasitpol P, Kaebnick H. Selective disassembly sequencing: a methodology for the disassembly of end-of-life products. *CIRP Annals-Manufacturing Technology* 2006;55:37-40.